

Главная > Блог > Web-разработка > Разное >

Расширение «Prettier — Code formatter» для VS Code

22.07.2022

Теги: IDE • JavaScript • Web-разработка • Конфигурация • Настройка • Плагин • Установка • Файл

Расширение позволяет форматировать код с помощью заданных в настройках правил. Позволяет разработчику сосредоточиться на написании кода, а не на его форматировании. При этом использует «усреднённый», наименее спорный подход к стилю при форматировании кода.

Установка расширения

После установки расширения нужно добавить в файл настроек VS Code `settings.json` новые опции. Можно установить форматирование для всех языков или только для некоторых.

// файл настроек VS Code settings.json

```
{
  .....
  "prettier.enable": true,
  "editor.formatOnSave": false, // форматировать код
  "editor.formatOnPaste": false, // форматировать пр
  "editor.defaultFormatter": null,
  "[javascript]": {
    "editor.formatOnSave": true,
    "editor.formatOnPaste": true,
    "editor.defaultFormatter": "esbenp.prettier-vs
  },
  "[markdown]": {
    "editor.formatOnSave": true,
    "editor.formatOnPaste": true,
    "editor.defaultFormatter": "esbenp.prettier-vs
  },
  .....
}
```

Есть несколько мест, где можно задать правила форматирования — это файл `.prettierrc`, файл `.editorconfig` или файл настроек VS Code `settings.json`. Если есть файл `.prettierrc` — настройки из `.editorconfig` и `settings.json` будут проигнорированы. Правила из `settings.json` — это запасной вариант и обычно предназначены только для использования с файлами, не относящимися к проекту.

Категории блога

- Web-разработка
- HTML и CSS
- JavaScript и jQuery
- PHP и MySQL
- CMS Битрикс
- CMS WoprdPress
- Yii2 и Laravel
- Python и Flask
- Web-аналитика
- Разное
- OC Linux
- Команды
- Сценарии
- Разное
- 1С:Предприятие
- Программирование
- Язык запросов
- Разное
- Локальная сеть
- Разное

Облако тегов

- 1С:Предприятие (31)
- Yii2 (69)
- API (29)
- БазаДанных (95)
- Bash (42)
- Битрикс (66)
- CLI (92)
- Блог (29)
- CMS (139)
- Верстка (43)
- CSS (50)
- ИнтернетМагаз... (84)
- Frontend (75)
- КаталогТоваров (87)
- HTML (66)
- Класс (29)
- JavaScript (143)
- Ключ (27)
- Laravel (72)
- Команда (63)
- Linux (135)
- Компонент (60)
- MySQL (75)
- Конфигурация (58)
- PHP (124)
- Корзина (32)
- React.js (66)
- ЛокальнаяСеть (28)
- Ubuntu (63)
- Модуль (34)
- Web-разработка (501)
- Навигация (31)
- WordPress (73)
- Настройка (133)
- ПанельУправле... (29)

```
// файл настроек VS Code settings.json
{
    .....
    // расширение «Prettier — Code formatter» для форматирования
    "prettier.arrowParens": "avoid",
    "prettier.bracketSpacing": true,
    "prettier.endOfLine": "lf",
    "prettier.htmlWhitespaceSensitivity": "css",
    "prettier.insertPragma": false,
    "prettier.jsxBracketSameLine": false,
    "prettier.jsxSingleQuote": false,
    "prettier.printWidth": 80,
    "prettier.proseWrap": "preserve",
    "prettier.quoteProps": "as-needed",
    "prettier.requirePragma": false,
    "prettier.semi": false,
    "prettier.singleQuote": true,
    "prettier.tabWidth": 4,
    "prettier.trailingComma": "es5",
    "prettier.useTabs": false,
    "prettier.vueIndentScriptAndStyle": false,
    "prettier.embeddedLanguageFormatting": "auto",
    .....
}
```

Плагин (33) Файл (44)
 Практика (99) Форма (58)
 Сервер (71) Фреймворк (192)
 Событие (27) Функция (36)
 Теория (97) Шаблон Сайта (68)
 Установка (63)

Все теги

Категории статей

Web-разработка
 1С:Предприятие
 ОС Linux
 Регулярные выражения
 Разное

Расширение будет искать файл конфигурации `.prettierrc` сначала в рабочей директории проекта, а в случае неудачи — во всех родительских директориях вплоть до корневой. Так что можно использовать один файл конфигурации для всех проектов.

```
[projects]
.prettierrc
[project-one]
  index.js
  package.json
  .....
[project-two]
  index.js
  package.json
  .....
```

Расширение будет использовать `prettier` из локальных зависимостей проекта (рекомендуется). Если `prettier.resolveGlobalModules:true` — будет использоваться глобальная установка (не рекомендуется). Если модуль не установлен ни локально, ни глобально — будет использоваться версия `prettier`, связанная с расширением.

```
$ npm install prettier --save-dev # локальная установка
```

Правила форматирования

Давайте создадим файл `.prettierrc` и установим для всех опций форматирования значения по умолчанию.

```
{
    "arrowParens": "always", // скобки вокруг единств
    "bracketSpacing": true, // пробелы между скобками
    "endOfLine": "lf", // окончания строк
```

```

"htmlWhitespaceSensitivity": "css", // форматирова
"insertPragma": false, // вставлять спец.комментар
"jsxBracketSameLine": false, // где будет завершак
"jsxSingleQuote": false, // использовать в jsx оди
"printWidth": 80, // на какой позиции выполнять пе
"proseWrap": "preserve", // как обрабатывать markd
"quoteProps": "as-needed", // свойства объекта в к
"requirePragma": false, // форматировать только фа
"semi": false, // точка с запятой в конце оператор
"singleQuote": false, // использовать одинарные ка
"tabWidth": 4, // заменять табуляцию на 4 пробела
"trailingComma": "es5", // конечные запятые в объе
"useTabs": false, // делать отступы с помощью табу
"vueIndentScriptAndStyle": false, // отступ внутри
"embeddedLanguageFormatting": "auto" // форматиров
}

```

arrowParens

Добавлять круглые скобки вокруг единственного параметра стрелочной функции. Возможные значения

- **always** — всегда добавлять скобки, например `(x) => x *`
- **avoid** — опускать скобки, когда это возможно, например `x => x * x`

bracketSpacing

Добавлять пробелы между скобками в литералах объектов. Возможные значения

- **true** — добавлять пробелы, например `{ foo: bar }`
- **false** — не добавлять пробелы, например `{foo: bar}`

endOfLine

Символы конца строки, возможные значения **lf**, **crlf** и **auto**.

Значение **auto** сохраняет существующие окончания строк (смешанные значения в одном файле нормализуются путем просмотра того, что используется в конце первой строки).

htmlWhitespaceSensitivity

Чувствительность к пробелам для html кода. Нельзя безопасно преобразовать первый фрагмент кода ниже во второй фрагмент, так как это может изменить отображаемый вывод в браузере.

Нужно учитывать, какой это элемент — строчный (**display:inline**) или блочный (**display:block**).

```
<a href="https://prettier.io/">Prettier is an opinionated
```

```
<a href="https://prettier.io/">
  Prettier is an opinionated code formatter.
</a>
```

Возможные значения

- **css** — учитывать CSS-свойство **display** для безопасного форматирования
- **strict** — пробелы (или их отсутствие) вокруг всех тегов считаются значительными
- **ignore** — пробелы (или их отсутствие) вокруг всех тегов считаются незначительными

insertPragma

Добавлять в начало файла комментарий со специальным маркером **@format**, который указывает, что файл был отформатирован. Возможные значения **true** или **false**.

```
/** @format */
```

[Копировать](#)

```
import { useState, useEffect } from 'react'  
import { useLocation } from 'react-router-dom'
```

jsxBracketSameLine

Разместить символ **>** многострочного html или jsx кода в конце последней строки или на следующей строке (не относится к самозакрывающимся элементам).

```
// jsxBracketSameLine = true  
<button  
  className="prettier"  
  id="prettier"  
  onClick={this.handleClick}>  
  Click Here  
</button>
```

[Копировать](#)

```
// jsxBracketSameLine = false  
<button  
  className="prettier"  
  id="prettier"  
  onClick={this.handleClick}  
>  
  Click Here  
</button>
```

[Копировать](#)

jsxSingleQuote

Заменять в jsx-коде двойные кавычки на одинарные, возможные значения **true** или **false**.

```
// jsxSingleQuote = true  
export default function Preloader() {  
  return <div className='preloader'></div>  
}
```

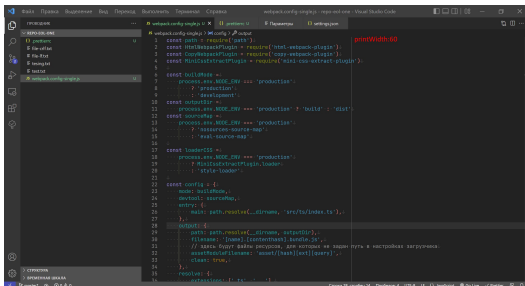
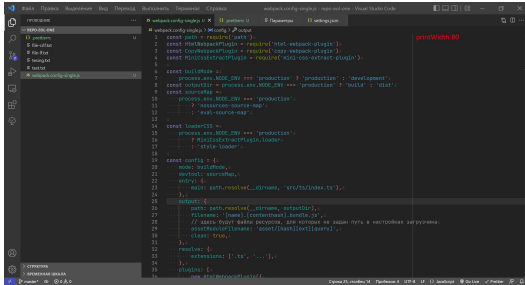
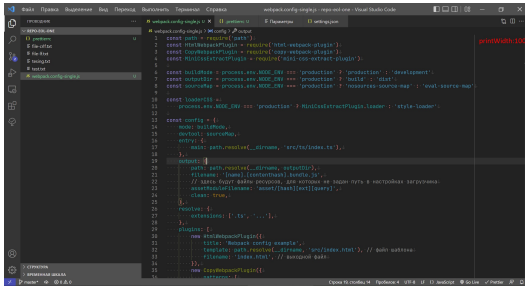
[Копировать](#)

```
// jsxSingleQuote = false  
export default function Preloader() {  
  return <div className="preloader"></div>  
}
```

[Копировать](#)

printWidth

Форматировать код таким образом, чтобы не выходить за ограничительную линию справа (насколько это возможно).



proseWrap

По умолчанию prettier не меняет markdown-текст, поскольку некоторые сервисы используют рендерер, чувствительный к разрыву строки. Чтобы разбить все блоки markdown-текста, когда они превышают ширину печати — следует использовать значение **always**. Чтобы убрать переносы строк из каждого блока markdown-текста — следует использовать значение **never**.

- **always** — разбить блок текста, если он превышает ширину печати
- **never** — развернуть каждый блок текста в одну длинную строку
- **preserve** — ничего не делать, оставить markdown-текст как есть

quoteProps

Добавлять или не добавлять кавычки для свойств объекта, возможные значения

- **as-needed** — добавлять кавычки для свойств только там, где это необходимо
- **consistent** — если хотя бы одно свойство требует кавычек, добавить для всех
- **preserve** — оставлять без изменений (но с заменой кавычек, см. **singleQuote**)

```
// код до форматирования
const settings = {
```

Копировать

```
tabWidth: 4,  
"use-tabs": false,  
}
```

```
// quoteProps = as-needed (singleQuote:true)  
const settings = {  
  tabWidth: 4,  
  'use-tabs': false,  
}
```

Копировать

```
// quoteProps = consistent (singleQuote:true)  
const settings = {  
  'tabWidth': 4,  
  'use-tabs': false,  
}
```

Копировать

```
// quoteProps = preserve (singleQuote:true)  
const settings = {  
  tabWidth: 4,  
  'use-tabs': false,  
}
```

Копировать

requirePragma

Форматировать только те файлы, которые имеют в начале комментариев со специальным маркером `@format`. Возможные значения `true` или `false`.

semi

Добавлять или нет точку с запятой в конце операторов, возможные значения `true` или `false`.

```
// semi = true  
import React from 'react';  
import ReactDOM from 'react-dom';
```

Копировать

```
// semi = false  
import React from 'react'  
import ReactDOM from 'react-dom'
```

Копировать

singleQuote

Заменять или нет двойные кавычки на одинарные, возможные значения `true` или `false`.

```
// singleQuote = true  
import React from 'react';  
import ReactDOM from 'react-dom';
```

Копировать

```
// singleQuote = false  
import React from "react";  
import ReactDOM from "react-dom";
```

Копировать

tabWidth

Заменять табуляцию на `tabWidth` пробелов, если `useTabs` равно `false`.

trailingComma

Добавлять или нет конечную запятую в массивах и объектах, возможные значения:

- `es5` — добавлять запятые в конце, если они допустимы в ES5 (объекты, массивы и т.д.)
- `none` — не добавлять запятые в конце
- `all` — добавлять запятые везде, где это возможно (включая параметры функций и вызовы)

```
// trailingComma = es5
const settings = {
  tabWidth: 4,
  useTabs: false,
}
```

[Копировать](#)

```
// trailingComma = none
const settings = {
  tabWidth: 4,
  useTabs: false
}
```

[Копировать](#)

useTabs

Делать отступы в коде с помощью табуляции, возможные значения `true` или `false`.

vueIndentScriptAndStyle

Добавлять или нет отступ для кода внутри тегов `<script>` и `<style>` в файлах Vue, возможные значения `true` или `false`.

embeddedLanguageFormatting

Форматировать или нет встроенный код, возможные значения

- `auto` — форматировать встроенный код, если удалось его идентифицировать
- `off` — никогда не форматировать встроенный код автоматически

Дополнительные возможности

Для форматирования других языков программирования, например PHP, нужно установить плагин для Prettier — тогда расширение «Prettier — Code formatter» подхватит этот плагин.

```
$ npm install @prettier/plugin-php --save-dev
```

[Копировать](#)

Файл настроек VS Code `settings.json`:

```
{
  .....
  "prettier.enable": true, // разрешить работу расши
  "editor.formatOnSave": false, // форматировать код
  "editor.formatOnPaste": false, // форматировать пр
```

[Копировать](#)

```

"editor.defaultFormatter": null,
"[javascript]": {
  "editor.formatOnSave": true,
  "editor.formatOnPaste": true,
  "editor.defaultFormatter": "esbenp.prettier-vs
},
"[php]": {
  "editor.formatOnSave": true,
  "editor.formatOnPaste": true,
  "editor.defaultFormatter": "esbenp.prettier-vs
},
.....
}

```

Файл опций форматирования `.prettierrc`:

```

{
  .....
  "phpVersion": "7.1",
  "trailingCommaPHP": true,
  "braceStyle": "psr-2"
}

```

Копировать

Но вообще, для работы с PHP есть более подходящие расширения, см. [здесь](#).

Вместо заключения

Не для всех файлов с кодом нужно использовать внешний форматтер, потому что у VS Code есть несколько встроенных. И зачастую они ничем не хуже внешних, а порой даже лучше. Так что для одних файлов можно использовать встроенные средства форматирования, а для других — использовать внешний форматтер.

```

"liveServer.settings.fullReload": false,
"liveServer.settings.port": 5555,
"liveServer.settings.donotShowInfoMsg": true,
"liveServer.settings.donotVerifyTags": true,
// Расширение Prettier для форматирования кода
"prettier.arrowParens": "avoid", // скобки вокруг
"prettier.bracketSpacing": true, // пробелы между
"prettier.endOfLine": "lf", // окончания строк
"prettier.htmlWhitespaceSensitivity": "css", //
"prettier.insertPragma": false, // вставлять спе
"prettier.jsxBracketSameLine": false, // где буд
"prettier.jsxSingleQuote": false, // использовать
"prettier.printWidth": 100, // на какой позиции
"prettier.proseWrap": "preserve", // как обработ
"prettier.quoteProps": "as-needed", // свойства
"prettier.requirePragma": false, // форматироват
"prettier.semi": true, // точка с запятой в конц
"prettier.singleQuote": true, // использовать од
"prettier.tabWidth": 4, // заменять табуляцию на
"prettier.trailingComma": "es5", // конечные зап
"prettier.useTabs": false, // делать отступы с п
"prettier.vueIndentScriptAndStyle": false, // от
"prettier.embeddedLanguageFormatting": "auto" //

```


Похожие записи

- Расширение «ESLint» для VS Code, часть 2 из 2
- Расширение «ESLint» для VS Code, часть 1 из 2
- Настройка PhpStorm для работы с Laravel
- Расширение «Remote — SSH» для VS Code
- Git. Символы конца строки EOL
- Webpack. Начало работы, часть 2 из 2
- Webpack. Начало работы, часть 1 из 2

Поиск: IDE • JavaScript • Web-разработка • Конфигурация • Настройка •
Плагин • Установка • Файл • Prettier

Узелки на память: Web-разработка, 1С:Предприятие, ОС Linux